

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6144845号
(P6144845)

(45) 発行日 平成29年6月7日(2017.6.7)

(24) 登録日 平成29年5月19日(2017.5.19)

(51) Int.Cl.	F I
B 2 5 J 13/00 (2006.01)	B 2 5 J 13/00 Z
B 2 5 J 9/22 (2006.01)	B 2 5 J 9/22 A

請求項の数 12 (全 20 頁)

(21) 出願番号	特願2016-549576 (P2016-549576)	(73) 特許権者	516227939
(86) (22) 出願日	平成26年2月5日(2014.2.5)		アーベーベール シュヴァイツ アクツイエ ンゲゼルシャフト
(65) 公表番号	特表2017-505239 (P2017-505239A)		スイス国 5400 バーデン, プロ ボヴェリ シュトラーセ 6
(43) 公表日	平成29年2月16日(2017.2.16)	(74) 代理人	110002077
(86) 国際出願番号	PCT/EP2014/052180		園田・小林特許業務法人
(87) 国際公開番号	W02015/117641	(72) 発明者	ゴスタウソン, マウヌス
(87) 国際公開日	平成27年8月13日(2015.8.13)		スウェーデン国 エス-43953 オー サ, ステークツレヴェーゲン 21
審査請求日	平成28年12月22日(2016.12.22)	(72) 発明者	ルース, マーク
早期審査対象出願			アメリカ合衆国 ミシガン 48044, マコーム, ミドル リバー ドライブ 50501

最終頁に続く

(54) 【発明の名称】 ショーを協調的に演じている複数のロボットの動きを定義するためのシステム及び方法

(57) 【特許請求の範囲】

【請求項1】

ショーを協調的に演じている複数の実際のロボットの動きを定義するプレイリストを作成し修正するためのシステムであって、

- 前記ショーの間の前記ロボットの動きを記述し且つ各セグメントが定義された期間における前記ロボットの内の1つのロボットの動きの記述であって、前記ロボットの位置を含む記述を含む複数のセグメントを保存するため、及び各シーンが前記ロボットによって同時に実行されるべき複数のセグメントを指定する複数のシーンを保存するための少なくとも1つの記憶装置(3、5)と、

- シーンのシーケンスを含むプレイリストの作成を処理し、ユーザコマンドに従って前記プレイリストを修正するように構成されたプレイリストハンドラー(6)と、

- ユーザが前記シーンのシーケンスを修正することによって前記プレイリストを編集することができるようにするために、前記ユーザに前記プレイリストを表示し、前記ユーザコマンドを受信するように構成されたユーザインターフェース(8)とを備え、

前記プレイリストハンドラー(6)は、前記シーンのシーケンスの一部の修正に関するユーザコマンドを受信して、前記プレイリストの前記修正された部分の前記シーンに含まれる前記セグメントの前記ロボットの動きの前記記述に基づいて前記修正を認証し、前記認証の結果に基づいて前記修正を許可する又は拒否するように構成され、前記プレイリストの前記修正された部分に隣接するシーンで同じロボットに属する位置と比較し、前記比較の結果に基づいて前記修正を認証するように構成されていることを特徴とする、システム

10

20

。

【請求項 2】

各セグメントが前記ロボットの開始位置と終了位置を含み、前記プレイリストハンドラー(6)は、前記プレイリストの前記修正された部分に隣接するシーンで同じロボットに属するセグメントの開始位置と終了位置とを比較することによって、修正を認証し、隣接するシーンの任意のセグメントの開始位置と終了位置との間の差が限界値を超えた場合に、前記修正を拒否するように構成されている、請求項 1 に記載のシステム。

【請求項 3】

前記保存されたセグメントに基づいて複数のシーンを作成し、前記記憶装置(5)に前記シーンを保存するように構成されたシーンハンドラー(4)を備える、請求項 1 又は 2 に記載のシステム。

10

【請求項 4】

前記セグメントの内の各セグメントは、前記セグメントがどのシーンに属するかの情報を含み、前記シーンハンドラー(4)は、前記セグメントがどのシーンに属するかの情報に基づいて前記セグメントを複数のシーンにグループ化するように構成されている、請求項 3 に記載のシステム。

【請求項 5】

前記セグメントの各セグメントは、前記セグメントがどのロボットに属するかの情報を含むロボット識別タグを含み、前記プレイリストハンドラー(6)は、前記ロボット識別タグに基づいてセグメントがどのロボットに属するかを決定するように構成されている、請求項 1 から 4 のいずれか一項に記載のシステム。

20

【請求項 6】

前記プレイリストハンドラー(6)は、前記プレイリスト、前記保存されたシーン、及び前記セグメントに基づいて前記ロボットの動きをシミュレーションし、前記シミュレーションに基づいて前記プレイリストを認証するように構成されている、請求項 1 から 5 のいずれか一項に記載のシステム。

【請求項 7】

各セグメントは更に、前記定義された期間の間に前記ロボットの動きを実行するためのプログラムコードを含み、前記システムは更に、前記プレイリスト、前記保存されたシーン、及び前記保存されたセグメントに基づいて前記ロボットの内の各ロボットのためのメインプログラムを生成するように構成されているプログラムジェネレータ(20)を備える、請求項 1 から 6 のいずれか一項に記載のシステム。

30

【請求項 8】

各セグメントは、前記ロボットコードを含み且つメインプログラムによってコールされるべきサブルーチン名を有するサブルーチンを含み、前記プログラムジェネレータ(20)は、前記プレイリストに基づいて前記サブルーチンへのコールのシーケンスを含む前記ロボットの各ロボットのためのメインプログラムを生成するように構成されている、請求項 7 に記載のシステム。

【請求項 9】

ショーを協調的に演じている複数の実際のロボットの動きを定義するプレイリストを作成し修正するための方法であって、

40

- 前記ショーの間の前記ロボットの動きを記述し、各セグメントが定義された期間の間の前記ロボットの内の 1 つのロボットの前記動きの記述であって、前記ロボットの複数の位置を含む記述を含む複数のセグメントを取得して保存する(40)ことと、

- 各シーンが、前記ロボットによって同時に実行されるべき前記複数のセグメントを含む複数のシーンを取得する(42)ことと、

- 前記シーンのシーケンスを含むプレイリストを作成する(44)こととを含み、前記方法は更に、

- ユーザに前記プレイリストを表示し、前記シーンのシーケンスを修正することによって、前記ユーザが前記プレイリストを編集することを可能にする(46)ことと、

50

- 前記シーンのシーケンスの一部の修正に関するユーザコマンドを受信する(48)ことと、
 - 前記プレイリストの前記修正された部分に隣接するシーンで同じロボットに属する前記位置の内の少なくとも幾つかを比較することと、
 - 前記比較の結果に基づいて前記修正を認証することと
 - 前記認証の結果に基づいて前記修正を許可する又は拒否することと、
 - 前記修正が許可された場合に、前記ユーザコマンドに基づいて前記プレイリストを修正する(54)ことと、
 - 前記修正されたプレイリストを保存することと
- を含むことを特徴とする方法。

10

【請求項10】

各セグメントは、前記ロボットの開始位置と終了位置を含み、前記方法は、前記プレイリストの前記修正された部分に隣接するシーンで同じロボットに属するセグメントの開始位置と終了位置とを比較することと、前記隣接するシーンの任意のセグメントの前記開始位置と終了位置との間の差が限界値を超えた場合に、前記修正を拒否することとを含む、請求項9に記載の方法。

【請求項11】

前記セグメントがどのシーンに属するかの情報を各セグメントが含む前記複数のセグメントを受信することと、前記セグメントがどのシーンに属するかの情報に基づいて前記シーンを作成することと、前記シーンを保存することとを含む、請求項9又は10に記載の方法。

20

【請求項12】

各セグメントは更に、前記定義された期間の間に前記ロボットの動きを実行するためのプログラムコードを含み、前記方法は更に、前記プレイリスト、前記保存されたシーン及び前記保存されたセグメントに基づいて前記ロボットの内各ロボットのためのメインプログラムを生成する(64)こととを含む、請求項9から11のいずれか一項に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ショーを協調的に演じている複数の実際のロボットの動きを定義するためのシステム及び方法に関する。

30

【背景技術】

【0002】

ロボットは、産業界において反復作業を実施するために最も多く使用される。この場合、重要なのは作業が正確に確実に完了しているということである。ロボットが作業を実施しているのを見ることがどれほど楽しいかは重要ではない。

【0003】

既存の実施では、人間のプログラマーが各作業についてロボットをプログラミングする。作業に関連する重要なロボットの位置は、ロボットを直接位置決めし、その位置を保存することによって定義されうる。プログラマーはその後、保存された位置を参照するコードを書き、ロボットに所望の作業を実施するように命令する。

40

【0004】

しかしながら、近年、人々を楽しませるために、ロボットを利用して、ビデオのディスプレイ、ビデオプロジェクター、又は階段等の娯楽機器を動かすことが流行している。例えば、ショーのステージ上で人間と共に演技しているロボットが多数いる。これには、人とロボットが出演しているミュージカルのショー、又はロボットがディスプレイパネルを動かすトレードショーがありうる。またこれは、ロボットが人と共にディスプレイやスキャン機器を操作するデザイナーの店やエリアにも適用できる。上記システムの課題は、ショーのプレイリスト、及びステージ上の動作の一方、又は両方に対して、ロボットのプロ

50

グラムをどのように構築し、ロボットの動きを同期させるかである。

【0005】

文献US 8 3 5 6 7 0 4には、ビデオのディスプレイ、ビデオプロジェクター又は階段等のロボットにより制御された娯楽要素が開示されている。ロボットは、三次元において娯楽要素を動かす。

【0006】

観客を楽しませる目的でロボットをプログラミングすることは、産業界においてロボットをプログラミングするのとは多くの点において質的に異なっている。ロボットは通常、見た目のためにデザインされた特定の動きのシーケンスの明確に定義されたセットを有する。ロボットの側に知的な意思決定は必要なく、見た目のみが重要視される。従来のプログラミング法を用いて、ロボットに自然な「有機的」に見える動きをさせるようにプログラミングすることは、よくても非常に困難で時間がかかることであり、最悪にはほぼ不可能である。

10

【0007】

プログラムには意思決定がなく、見た目が重要視されるため、楽しませる演技のためのロボットの動きのプログラミングは、従来のロボットのプログラミングよりもコンピュータのアニメーションと共通点を持つ。そうであれば、理想的な解決策は、コンピュータのアニメーターがロボットの演技のアニメを定義すると全く同じ方法でロボットの演技を定義することである。

【0008】

20

既存のコンピュータの3Dアニメーションソフトウェアと、既存のコンピュータのアニメーション制作能力とを用いて、ロボットのアニメーションシーケンスを製作することは周知である。コンピュータのアニメーターはコンピュータのアニメーションソフトウェアを使用し、実際のロボットの物理的な限界内で実際のロボットのモデルをアニメ化する。次に、実際のロボットの動きがアニメ化されたロボットの動きとぴったり一致するように、アニメーションからロボットのプログラムを製作する。

【0009】

今日、ショーデザイナーは外部の3Dアニメーションツール、例えばMAYAを使って、人間の動きと共にロボットの動き、及びショー、又はトレードショーにおける音楽同期を設計する。ショーデザイナーは、ショーが演じられるべき場所から遠く離れたスタジオで、ロボットのソフトウェアモデルを使用して、ロボットの動きを予め設計する。ショーが、実際のステージ上で実際のロボットが演じ、実際に人がショーで演技をしている時に、ロボットの動きを、ロボットアニメーションのシーケンスを製作中には予測できなかった実際の環境、及びその他の状況に適合させるために変える必要が多く発生する。このため、アニメーションと、ロボットのプログラムコードが製作された後で、ロボットの動きを編集する必要がある。特に、例えばショーのリハーサルの間等に、実際のロボットと共に現場でロボットの動きを編集する必要がある。例えば、ロボットの動きのシーケンスを変える必要がある。しかしながら、ロボットの動きは、ロボットのアニメーションシーケンスの製作中に、時間整合されていた。このため、ロボットの動きのシーケンスを変える時に、ロボット同士が衝突する危険性がある。

30

40

【発明の概要】

【発明が解決しようとする課題】

【0010】

本発明の目的は、コンピュータのアニメーションソフトウェアによって定義されるロボットの動きのシーケンスの編集に関する上述の問題に解決策を提供することである。

【課題を解決するための手段】

【0011】

本発明の一態様によれば、この目的は請求項1によって定義されるように、システムによって達成される。

【0012】

50

本システムは、ショーの間のロボットの動きを記述し且つ各セグメントが定義された期間におけるロボットの内の1つのロボットの動きの記述を含む複数のセグメントを保存するため、及び各シーンがロボットによって同時に実行されるべき複数のセグメントを指定する複数のシーンを保存するための少なくとも1つの記憶装置と、シーンのシーケンスを含むプレイリストの作成を処理し、ユーザコマンドに従ってプレイリストを修正するように構成されたプレイリストハンドラーと、ユーザがシーンのシーケンスを修正することによってプレイリストを編集することができるようにするために、ユーザにプレイリストを表示し、ユーザコマンドを受信するように構成されたユーザインターフェースとを備える。

【0013】

ロボットのために定義されたショーの間に実行されるべき動きは、複数のシーンに分割される。各ロボットの動きは、複数の連続的なセグメントに分割される。各セグメントは、定義された期間の間に特定のロボットによって実行されるべき動きを定義する。定義された期間は、実行されるべき動きの継続時間を指定するのみであり、ショーの間いつセグメントが実行されるべきかを指定するものではない。各シーンに対し、ロボットによって同時に実行されるべき動きを含む幾つかの対応するセグメントがある。同時に実行されるべきセグメントの数は、ロボットの数と同じである。従って、各シーンのセグメントの数は、プログラミングされるべきロボットの数と等しい。各シーンは、どのセグメントがそのシーンに属するかの情報を含む。

【0014】

**本発明によれば、本システムは複数のシーンを取得し、各シーンは、ロボットによって同時に実行されるべき複数のセグメントを含む。シーンは、ロボットによって同時に実行されるべき動きを含む全てのセグメントを含む。プレイリストハンドラーは、ショーの間のロボットの動きのシーケンスを定義するシーンのシーケンスを含むプレイリストの作成を処理するように構成されている。本システムは、ユーザがシーンのシーケンスを修正することによってプレイリストを編集することができるようにするために、プレイリストをユーザに表示し、修正に関するユーザコマンドを受信するように構成されたユーザインターフェースを含む。プレイリストハンドラーは、ユーザコマンドに基づいて、プレイリスト、従ってシーンのシーケンスを修正するように構成されている。このため、ユーザがシーンのシーケンスを変更する、すなわち、シーン全体を移動する、追加する及び/又は削除することが可能になる。ロボットのアニメーションシーケンスが製作された時に、シーン内のロボットの動きは互いに調整され、衝突に対してチェックされているので、ユーザは修正に起因するロボット同士の衝突の危険性を伴わずにシーンのシーケンスを編集することができる。本発明により、ユーザが、ロボットのプログラムが各ロボットコントローラに配布される前に、コンピュータのアニメーションソフトウェアによって定義されたロボットの動きのシーケンスを編集することが可能になる。

【0015】

本発明の一実施形態によれば、本システムは、ショーの間のロボットの動きを記述し、各セグメントが定義された期間におけるロボットの内の1つのロボットの動きの記述を含む複数のセグメントを保存するセグメント記憶装置を備える。

【0016】

本発明の一実施形態によれば、本システムは、シーンを保存するためのシーン記憶装置と、保存されたセグメントに基づいて複数のシーンを作成し、シーン記憶装置にそのシーンを保存するように構成されたシーンハンドラーとを備える。

【0017】

本発明の一実施形態によれば、プレイリストハンドラーは、シーンのシーケンスの一部の修正に関するユーザコマンドを受信して、プレイリストとの修正された部分のシーンに含まれるセグメントのロボットの動きの記述に基づいて修正を認証し、認証の結果に基づいて修正を可能にする又は拒否するように構成されている。一シーンにおけるロボットの位置は、ロボットが次のシーンにおいて移動できる位置に制限を課す。この実施形態は、

10

20

30

40

50

ロボットの動きの推奨される修正を実行することが可能か否か、すなわちロボットが新たな動きのシーケンスを演じることが可能か否かの認証、すなわちチェックを実行する。推奨される修正が実行不可能である場合、修正は拒否される。この実施形態は、ユーザが、ロボットが動きを演じることができないような動きのシーケンスの修正を行うことを防止する。

【0018】

本発明の一実施形態によれば、ロボットの動きの記述は、ロボットの位置を含み、プレイリストハンドラーは、プレイリストの修正された部分に隣接するシーンで同じロボットに属する位置を比較して、比較の結果に基づいて修正を認証するように構成されている。この位置は例えば、デカルト位置又はジョイント値として表すことができる。

10

【0019】

本発明の一実施形態によれば、各セグメントは、ロボットの開始位置と終了位置を含み、プレイリストハンドラーは、プレイリストの修正された部分に隣接するシーンで同じロボットに属するセグメントの開始位置と終了位置とを比較することによって修正を認証し、隣接するシーンの任意のセグメントの開始位置及び終了位置との間の差が限界値を超えた場合に、修正を拒否するように構成されている。シーンの一セグメントの終了位置と、次のシーンの同じロボットに属するセグメントの開始位置との間の差が限界値を超えた場合、修正は許可されない。

【0020】

この実施形態により、ロボットが動きを実行することができないように、ユーザの動きのシーケンスの修正が防止される。

20

【0021】

例えば、記述には、ロボットの軸の複数のジョイント値を含み、プレイリストハンドラーは、プレイリストの修正された部分に隣接するシーンで同じロボットに属するセグメントの軸のジョイント値を比較するように構成されている。

【0022】

本発明の一実施形態によれば、各セグメントは、セグメントがどのシーンに属するかの情報を含み、シーンハンドラーは、それに基づいてセグメントを複数のシーンにグループ化するように構成されている。各セグメントに、そのセグメントがどのシーンに属するかの情報を付与することで、シーンハンドラーのシーンの作成が促進される。

30

【0023】

本発明の一実施形態によれば、各セグメントは、セグメントがどのロボットに属するかの情報を含むロボット識別タグを含み、プレイリストハンドラーは、ロボット識別タグに基づいてセグメントがどのロボットに属するかを決定するように構成されている。この情報は例えば、各ロボットのメインプログラムを作成し、正しいロボットにメインプログラムを配布するために、ロボットの動きの修正を認証するために使用される。

【0024】

本発明の一実施形態によれば、プレイリストハンドラー及びユーザインターフェースは、ユーザがプレイリストのシーンのセグメントを編集することを可能にするように構成されている。選択されたシーンのセグメントを編集することができる。例えば、新たなセグメントを追加することができ、ライブラリからの既存のセグメントを追加することができ、セグメントをプレイリストから削除することができ、セグメントの順番を変更することができる。

40

【0025】

しかしながら、セグメントが変化すると、ロボット同士が衝突する危険性が生じうる。本発明の一実施形態によれば、プレイリストハンドラーはプレイリスト、保存されたシーン、及び保存されたセグメントに基づいてロボットの動きをシミュレーションし、シミュレーションに基づいてプレイリストを認証するように構成されている。この実施形態は、セグメントの変更が原因の衝突やエラーを避けるために、プレイリストによって定義されたロボットの動きのシーケンスを認証する。

50

【 0 0 2 6 】

本発明の一実施形態によれば、各セグメントは更に、定義された期間においてロボットの動きを実行するためのプログラムコードを含み、本システムは更に、プレイリスト、保存されたシーン、及び保存されたセグメントのロボットコードに基づいて各ロボットのメインプログラムを生成するように構成されたプログラムジェネレータを備える。この実施形態により、コンピュータのアニメーションソフトウェア、及びユーザによって作成されたプレイリストによって定義されたロボットの動きのシーケンスに基づいて個々の移動を実行するために、各ロボットに対してロボットプログラムを自動的に生成することが可能になる。

【 0 0 2 7 】

本発明の一実施形態によれば、各セグメントは、ロボットコードを含み、メインプログラムによってコールされるサブルーチン名を有するサブルーチンを含み、プログラムジェネレータは、プレイリストに基づいてサブルーチンへのコールのシーケンスを含む各ロボットのメインプログラムを生成するように構成される。メインプログラムは、移動を実行するためのいかなるロボットコードも含まない。したがって、移動を実行するためのロボットコードを含むセグメントを、メインプログラムとは別のロボットコントローラに読み込むことができる。ロボットコードをロボットコントローラに読み込むには時間がかかることが多い。この実施形態では、ロボットコードは1回だけロボットコントローラに読み込むだけでよい。ロボットプログラムはロボットコードを含むサブルーチンへのコールのシーケンスを含むため、ロボットプログラムをロボットコントローラに読み込むのは簡単で時間がかからない。ロボットの動きのシーケンスを後で修正すべき場合、メインプログラムのサブルーチン読み出しシーケンスを修正することは簡単であり、新たなメインプログラムをロボットコントローラに読み込むのは時間がかからない。ロボットコードは修正なしでもよい。この実施形態により、ロボットコードがロボットコントローラに読み込まれた後のロボットプログラムへの修正が簡単になる。

【 0 0 2 8 】

本発明の一実施形態によれば、本システムは、各ロボットがそのロボットに属する関連メインプログラムとセグメントを受信するように、各ロボットにメインプログラムとセグメントを自動的に配布するように適合される。シーンのシーケンスの編集と、ロボットのメインプログラムの生成は、ロボットコントローラとは離れたコンピュータでオフラインで行われることが好ましい。この実施形態により、プログラムコードを含むメインプログラムとセグメントの各ロボットへの自動的な配布が可能になる。

【 0 0 2 9 】

本発明の別の態様によれば、この目的は、請求項12によって定義され、アニメーションソフトウェアによって定義されるように、ショーの間のロボットの動きの記述に基づいて、ショーを協調的に演じている複数の実際のロボットの動きを定義するための方法によって達成される。

【 0 0 3 0 】

本方法は、

- アニメーションソフトウェアによって定義されるロボットの動きを記述し、各セグメントが定義された期間内のロボットの内の1つのロボットの動きの記述を含む複数のセグメントを取得してすることと、
- セグメントを保存することと、
- 各シーンが、ロボットによって同時に実行されるべき複数のセグメントを含む複数のシーンを取得することと、
- シーンについての情報を保存することと、
- シーンのシーケンスを含むプレイリストを作成することと、
- ユーザにプレイリストを表示し、シーンのシーケンスを修正することによって、ユーザがプレイリストを編集することを可能にすることと、
- 修正に関するユーザコマンドを受信することと、

- ユーザコマンドに基づいてプレイリストを修正することと、
- 修正されたプレイリストを保存することと

を含む。

【0031】

複数のセグメントは、ロボットのアニメーターを含むモーション作成モジュールからセグメントを受信する又は読み出すことによって、取得することができる。複数のシーンは、モーション作成モジュールからシーンを受信する又は読み出すことによって、又は保存されたセグメントに基づいてシーンを作成することによって取得することができる。

【0032】

本発明の一実施形態によれば、本方法は、

- シーンのシーケンスの一部分の修正に関するユーザコマンドを受信することと、
 - プレイリストの修正された部分のシーンに含まれたセグメントのロボットの動きの記述に基づいて修正を認証すること、そして、認証の結果に基づいて修正を許可する又は拒否すること、及び修正が許可された場合に、プレイリストを修正することと
- を含む。

【0033】

本発明の一実施形態によれば、ロボットの動きの記述はロボットの複数の位置を含み、本方法は、プレイリストの修正された部分において隣接するシーンで同じロボットに属する少なくとも幾つかの位置を比較し、比較の結果に基づいて修正を認証することを含む。

【0034】

本発明の一実施形態によれば、各セグメントは、ロボットのための開始位置と終了位置を含み、本方法は、プレイリストの修正された部分に隣接するシーンで同じロボットに属するセグメントの開始位置と終了位置とを比較することと、隣接するシーンの任意のセグメントの開始位置と終了位置との間の差が限界値を超えた場合に、修正を拒否することとを含む。

【0035】

本発明の一実施形態によれば、本方法は、複数のセグメントを受信することと、セグメントを保存することと、保存されたセグメントに基づいて複数のシーンを作成することと、シーンを保存することとを含む。

【0036】

本発明の一実施形態によれば、各セグメントは、セグメントがどのシーンに属するかの情報を含み、複数のシーンは、セグメントがどのシーンに属するかの情報に基づいて作成される。

【0037】

本発明の一実施形態によれば、各セグメントはセグメントがどのロボットに属するかの情報を含むロボット識別タグを含み、本方法は、ロボット識別タグに基づいてセグメントがどのロボットに属するかを決定することを含む。

【0038】

本発明の一実施形態によれば、本方法は更に、プレイリスト、保存されたシーン、及び保存されたセグメントに基づいてロボットの動きをシミュレーションすることと、シミュレーションに基づいてプレイリストを認証することとを含む。

【0039】

本発明の一実施形態によれば、各セグメントは更に、定義された期間においてロボットの動きを実行するためのプログラムコードを含み、本方法は更に、プレイリスト、保存されたシーン、及び保存されたセグメントのロボットコードに基づいて各ロボットのメインプログラムを生成することを含む。

【0040】

本発明の一実施形態によれば、各セグメントは、ロボットコードを含むサブルーチン、及びセグメントのロボットコードをコールするためのサブルーチン名を含み、本方法は、プレイリストに基づいてサブルーチンへのコールのシーケンスを含む各ロボットのメイン

10

20

30

40

50

プログラムを生成することを含む。

【0041】

本発明の一実施形態によれば、本方法は、各ロボットがそのロボットに属する関連メインプログラムとセグメントを受信するように、各ロボットにメインプログラムとセグメントを自動的に配布することを含む。

【0042】

ここで本発明を、本発明の様々な実施形態の記述と、添付の図面を参照することによって、より詳細に説明する。

【図面の簡単な説明】

【0043】

【図1】本発明の一実施形態に係るショーを協調的に演じている複数のロボットの動きを編集し、認証するためのシステムのブロックスキームを示す図である。

【図2】プレイリスト、シーン及びセグメントを保存するための構造の一例を示す図である。

【図3】本発明の一実施形態に係る、ショーを協調的に演じている複数のロボットのメインプログラムを生成するためのシステムのブロックスキームを示す図である。

【図4】ショーを協調的に演じている複数のディスプレイ処理ロボットを示す図である。

【図5】本発明の一実施形態に係る、ショーを協調的に演じている複数のロボットの動きを編集し、認証するための方法を示すフロー図である。

【図6】本発明の一実施形態に係る、ショーを協調的に演じている複数のロボットのメインプログラムを生成するための方法を示すフロー図である。

【発明を実施するための形態】

【0044】

図1に、本発明による、ショーを協調的に演じている複数のロボットの動きを編集し認証するためのシステムの一例のブロックスキームを示す。

【0045】

ロボットの動きは、ロボットアニメーションモジュール2aと、ロボットアニメーションモジュール2によって定義されたロボットの動きの記述に基づいてセグメントを作成するように適合されたセグメント作成モジュール2bとを備えるモーション作成モジュール1によって予め定義される。ロボットアニメーションモジュール2aと、セグメント作成モジュール2bは、本発明のシステムの一部ではない。

【0046】

ロボットアニメーションモジュール2aは、一又は複数のロボットのモデルに基づいてロボットをアニメ化するためのアニメーションソフトを含む。セグメント作成モジュール2bは、ロボットのモデルのアニメーションを分析し、実際のロボットにアニメ化されたロボットの動きを再生するように命令するロボットプログラムコードを自動的に生成するように適合されたコンピュータプログラムである。セグメント作成モジュール2b及びロボットアニメーションモジュール2aは、コンピュータ上で実行される。コンピュータプログラムはこれを、アニメ化されたロボットのモデルの回転コンポーネントを実際のロボットの軸に関連付けする手段と、完成したアニメーションからロボットのプログラムコードを生成する手段を提供することによって達成する。回転コンポーネントが実際のロボットの軸に関連付けられ、ロボットのモデルがアニメ化されると、プログラムは2つの段階：分析及び合成で機能することによってロボットのプログラムコードを生成する。分析段階では、アニメでのロボットの動きが決定される。次に合成段階において、分析段階から得られた情報を使用して、実際のロボットに同じように動くように命令するロボットのプログラムコードが生成される。

【0047】

分析段階では、コンピュータプログラムは、実際のロボットの軸の回転に関するコンピュータのアニメーションモデルの回転コンポーネントを分析する。コンピュータプログラムは、アニメーションの期間全体のこれらの回転を分析し、各軸の経路を決定する。コン

10

20

30

40

50

コンピュータプログラムは、回転値の線形変化を検出し、回転的な動きが変化するキーポジションを位置決めすることによってこの経路を決定する。コンピュータプログラムは次に、全ロボットのキーポジションのセットに個別の軸の動きの経路を組み合わせる。各位置は、各軸の回転値と、その位置の時間値とを含む。このキーポジションのセットは、実際のロボットに同じように動くように命令するプログラムを連続的に生成するのに使用可能な形態でアニメーションソフトによって定義されるように、ロボットの動きを完全に、また明確に記述する。

【 0 0 4 8 】

合成段階では、コンピュータプログラムは、分析段階で決定されたキーポジションのセットに踏み込み、ロボットのプログラミング言語で命令を生成する。結果的に得られたプログラムコードは、正確な時間にそれぞれ正確な位置に到達するようにロボットに命令する。これは、ロボットが前の位置に到達した時に実行されるべきそれぞれの移動命令を生成し、その位置に関連づけられる正確な時間にロボットに移動を終わらせ、キーポジションに到達させる移動命令に必要な正確な速度を計算することによって達成される。この結果、キーポジションを通るロボットの移動は、アニメ化されたロボットの移動と正確に一致する。あるいは、各移動時間は、特定の速度だけでなく、特定の時間でターゲット間を移動することができるこれらのロボット言語で使用するために保存される。

10

【 0 0 4 9 】

セグメント作成モジュール 2 b は、ロボットが定義された動きを演じるのに必要な全ての情報を含む複数のセグメントを作成するように構成されている。各セグメントは、定義された期間において特定のロボットによって実行されるべき動きを定義する。定義された期間は、例えば 1 0 分、あるいは 3 0 秒等の、実行されるべき動きの継続期間のみを指定する。定義された期間では、ショーの間にいつセグメントが実行されるべきかは指定されない。各ロボットによって実行されるべき動きのシーケンスは、複数の連続的なセグメントに分割される。各セグメントに対し、他のロボットによって同時に実行されるべき動きを含む幾つかの対応するセグメントがある。同時に実行されるべきセグメントの数は、ショーにおいて演じているロボットの数と等しい。

20

【 0 0 5 0 】

各セグメントは、ショーの定義された期間において実行されるべきロボットの動きの記述を含む。記述は、期間と、その期間におけるロボットの位置を含む。例えば、ロボットの位置は、ロボットの軸に対する角度位置のシーケンスを含みうる。あるいは、ロボットの位置は、ロボットのデカルト位置のシーケンスを含みうる。各セグメントは、ロボットの開始位置、複数の中間位置、及びロボットの終了位置を含みうる。

30

【 0 0 5 1 】

更に、各セグメントはセグメントがどのロボットに属するかの情報を含む。例えば、各セグメントは、ロボット ID の情報を有するロボット識別タグを含む。この実施形態では、一シーンごとのロボットに対するセグメントは 1 つのみである。各セグメントはまた、セグメントがどのシーンに属するかを識別することが可能になる情報も含むべきである。例えば、各セグメントはセグメントが属するシーンを識別するシーン識別タグを含みうる。本発明の一実施形態では、各セグメントはまた、セグメントの定義された期間においてロボットの動きを実行するための命令を含むロボットプログラムコードも含む。例えば、各セグメントは、ロボットコードを含むサブルーチンと、サブルーチンをコールするための一意な名前を含む。各セグメントに一意な名前が割り当てられていることが好ましい。

40

【 0 0 5 2 】

本発明によるシステムの一実施形態は、セグメント作成モジュール 2 b によって作成されたセグメントを保存するためのセグメント記憶装置 3、保存されたセグメントに基づいて複数のシーンを作成するように構成されたシーンハンドラー 4、シーンを保存するためのシーン記憶装置 5、シーンのシーケンスを含むプレイリストを作成するように構成されたプレイリストハンドラー 6、プレイリストを保存するためのプレイリスト記憶装置 7、及びシーンのシーケンスを修正することによってユーザがプレイリストを修正することを

50

可能にするために、ユーザにプレイリストを表示し、修正に関するユーザコマンドを受信するように構成されたユーザインターフェース 8 を含む。本システムは、コンピュータ上で実行するのに好適である。セグメント記憶装置 3、シーン記憶装置 5、及びプレイリスト記憶装置 7 は、例えば従来コンピュータで使用される RAM、ROM、及び PROM 等のいかなる種類のメモリであってもよい。これらを個別のデータ記憶装置、又は共通メモリの一部として実装することができる。シーンハンドラー 4 及びプレイリストハンドラー 6 は、データプロセッサ手段、例えば中央処理ユニット (CPU) 等を含むコンピュータ上で実行されるソフトウェアモジュールとして実装可能である。本システムは個別のコンピュータを含むことができる、あるいはセグメント作成モジュール 2 b 及びロボットアニメーションモジュール 2 と同じコンピュータを使用することができる。ユーザインターフェース 8 は例えば、ディスプレイスクリーン、及び例えばキーパッド、データマウス、又はタッチスクリーン等の従来の任意のユーザ入力手段によって実装することができる。

【0053】

シーンハンドラー 4 は、セグメントを複数のシーンにグループ化するように構成されている。一シーンは、ロボットによって同時に実行されるべき全てのセグメントを含む。このため、各シーンのセグメントの数は、ショーで演じているロボットの数と同じである。例えば、シーンハンドラー 4 はセグメントのシーン識別タグを使用して、どのセグメントがどのシーンに属するかを決定する。各シーンに対し、どのセグメントがそのシーンに属するかが決定される。シーンハンドラー 4 は、そのシーンをシーン記憶装置 5 に保存する。保存された各シーンは、どのセグメントがそのシーンに属するかの情報を含む。シーンは、セグメント自体を含みうる、あるいは、セグメント記憶装置のセグメントへのリンク / アドレスを含みうる。

【0054】

一実施形態では、シーン記憶装置 5 は、シーンを表す複数のアレイを含み、以下においてシーンアレイと称する。各シーンアレイは 1 つのシーンに対応し、セグメント記憶装置においてシーンに属するセグメントを指し示す複数の索引を含む。例えば、シーンアレイは、第 1 のインデックスがシーンのインデックス、例えばシーン識別タグであり、第 2 のインデックスがセグメントのインデックスである二次元アレイである。各シーンアレイのセグメントのインデックスの数は、ショーにおけるロボットの数と同じである。

【0055】

プレイリストハンドラー 6 は、シーンのシーケンスを含むプレイリストを作成し保存するように構成されている。シーン識別タグは、シーンの順番の情報を含むことが適切である。例えば、シーン情報タグは、シーンの順番を識別する数を含む。一実施形態では、プレイリストハンドラー 6 はシーン記憶装置 5 からシーンについての情報を読み出して、シーン識別タグに基づいてシーンのシーケンスを含むデフォルトのプレイリストを作成する。プレイリストについての情報は、プレイリスト記憶装置 7 に保存される。一実施形態では、プレイリストは、シーン記憶装置 5 のシーンに対応するシーンアレイを指し示す複数のインデックスを含むアレイである。以下において、アレイはプレイリストアレイと称される。

【0056】

図 2 に、プレイリスト、シーン、及びセグメントを保存するための構造の一例を示す。セグメント 1 3 は、セグメント記憶装置 3 に保存される。シーンは、複数のシーンアレイ 1 2 を含むシーン記憶装置 5 に保存される。シーンアレイ 1 2 の数は、保存されたシーンの数と同じである。各シーンアレイ 1 2 は、セグメント記憶装置 3 においてシーンに属するセグメント 1 3 を指し示す複数のインデックス 1 5 を含む。各シーンアレイ 1 2 のインデックス 1 5 の数は、ショーにおけるロボットの数と同じである。図 2 に示す例において、ロボットの数は 6 であり、従って各シーンアレイ 1 2 はセグメント記憶装置 3 において 6 つのセグメントを指し示す。プレイリストは、プレイリストにおいてシーンに対応するシーンアレイ 1 2 を指し示す複数のインデックス 1 4 からなる少なくとも 1 つのプレイリストアレイ 1 0 を含むプレイリスト記憶装置 7 に保存される。プレイリストの記憶装置は

10

20

30

40

50

、異なるプレイリストに対応する複数のプレイリストアレイを含みうる。この構造は、プレイリストの修正を簡略化して、データとプログラムコードの各ロボットへの読み込みを促進するため有利である。しかしながら、プレイリスト、シーン及びセグメントを記憶するための他の構造を使用することが可能である。一実施形態では、セグメントをシーンと共に共通の記憶装置に保存することができる。

【0057】

プレイリストハンドラー6は、ユーザインターフェース8上にプレイリストを表示するように構成されている。ユーザインターフェースはプレイリストを表示して、ユーザのプレイリストの編集を可能にするように設計されている。ユーザは、ユーザインターフェース上に表示されたプレイリストを編集する、例えば、シーンの順番を変更する、シーンを削除する、プレイリストに新たなシーンを追加することができる。しかしながら、ユーザは個別のセグメントを修正することはできない。ユーザインターフェース8は、プレイリストの修正に関するユーザからの入力に基づいて、コマンドを生成し、コマンドをプレイリストハンドラー6へ送るよう構成されている。プレイリストハンドラー6は、ユーザによって発行されたプレイリストの修正に関するユーザインターフェース8からのコマンドを受信し、そのコマンドに基づいてプレイリストを修正するように構成されている。修正されたプレイリストは、ユーザインターフェース8上に表示される。プレイリストハンドラー6は、ユーザが表示し編集すべき保存されたプレイリストのいずれかを選択することが可能になるように、2以上のプレイリストを保存するように構成されている。

【0058】

しかしながら、ロボットがあるシーンの特定位置へ移動した場合、次のシーンで実施できるロボットの動きが制限される。例えば、ロボットがあるシーンの特定位置に移動した場合、次のシーンの動きは、前のシーンの最後の位置の近くで開始しなければならない。ロボットのいずれかの軸に対する角度位置が、あるシーンの終了位置と次のシーンの開始位置との間で違いすぎる場合、あるシーンから次のシーンまでロボットを移動させるのは不可能でありうる。従って、あるシーンのロボットの軸の位置により、次のシーンでロボットの軸を移動させることができる位置に制限が課される。本発明の一実施形態では、プレイリストハンドラー6は、修正が実行される場目に修正を認証するように構成されている。プレイリストのシーンシーケンスの修正に関するコマンドを受信すると、プレイリストハンドラー6は、各ロボットについて、プレイリストの修正部分の隣接するシーンのセグメントに含まれる少なくとも幾つかの位置を比較して、比較結果に基づいて修正を認証するように構成されている。例えば、プレイリストハンドラーは、プレイリストの修正部分の隣接するシーンの同じロボットに属するセグメントの開始位置及び終了位置を比較して、隣接するシーンのいずれかのセグメントの開始位置と終了位置との差が限界値を超えた場合に、修正を拒否するように構成されている。例えば、認証を合格するためには、前のセグメントの最後のジョイント値が、次に続くセグメントの最初のジョイント値と等しい、又は非常に近いものでなければならない。

【0059】

例えば、あるシーンをプレイリストから除去すべきである場合、プレイリストハンドラーは、各ロボットについて、前のシーンのセグメントの終了位置を、新たなシーンのシーケンスにおける次のシーンのセグメントの開始位置と比較する。プレイリストハンドラーは、全てのロボットについて、隣接するシーンのセグメントの開始位置と終了位置との差が限界値未満である場合に、修正を許可する。プレイリストハンドラーは、いずれかのロボットの新たなシーンシーケンスにおける隣接するシーンの開始位置と終了位置との差が限界値を超えた場合、修正を拒否する。位置がロボットの軸に対してジョイント角である場合、各ロボットについて、前のシーンのセグメントの終了ジョイント角が、次のシーンのセグメントの対応する開始ジョイント角と比較される。

【0060】

任意選択的に、プレイリストハンドラーは、プレイリスト、保存されたシーン及びセグメントに基づいてロボットの動きをシミュレーションするように構成されたシミュレーシ

10

20

30

40

50

ョンユニットを含むことができ、プレイリストハンドラーは、このシミュレーションに基づいてプレイリストを認証するように構成されている。ユーザがプレイリストを編集し終わると、プレイリストハンドラーは、最終ステップとして、プレイリストによって決定された順番にシーンをシミュレーションし、ロボット間にコリジョンがないことをチェックする。プレイリストハンドラーが、いかなるロボット間のコリジョンを検出した場合、ユーザに警告が表示される。

【 0 0 6 1 】

ユーザがプレイリストを編集し終わり、プレイリストが認証され、コリジョンが見つけれなければ、ロボットのロボットプログラムが生成されるべきである。

【 0 0 6 2 】

図3に、本発明の一実施形態による、ショーを強制的に演じている複数のロボットのメインプログラムを生成するためのシステムのブロックスキームを示す。この実施形態では、各セグメントは、定義された期間においてロボットの定義された動きを実行するためのプログラムコードを含む。各セグメントのロボットコードは、メインプログラムからコールされるべき名前を有するサブルーチンを含む。サブルーチンは、セグメントに対して定義された動き、すなわちアニメーションソフトウェアによってセグメントに対して定義される動きを演じさせるために、ロボットに命令するためのプログラム命令を含む。本システムは、プレイリスト、保存されたシーン及びセグメントに基づいて、各ロボットのメインプログラムを生成するように構成されているプログラムジェネレータ20を備える。この実施形態では、各ロボットは、ロボットのメインプログラムを実行するためのハードウェア及びソフトウェアを含むロボットコントローラ22を有する。ロボットコントローラは、図3においてRC1~RCnとして図示されている。ユーザインターフェースは、ユーザが、選択されたプレイリストに基づいてメインプログラムの生成を指図し、またメインプログラムをそれぞれのロボットコントローラへ読み込むように指図することができるように構成されている。

【 0 0 6 3 】

プログラムジェネレータ20は、各ロボット用のメインプログラムを生成するように構成されている。各メインプログラムは、メインプログラムの実行中にコールされるべきサブルーチン名のシーケンスを含む。サブルーチン名のシーケンスは、プレイリストのシーンのシーケンスによって決定される。例えば、ロボットのメインプログラムの生成は、プレイリストの各シーンに対して、ロボットに対応するセグメントのサブルーチン名を検索することによって実行される。プログラムジェネレータ20はまた、ロボットに属する全てのセグメントを含む各ロボットのセグメントファイル、又は少なくともセグメントからの情報を生成するようにも構成される。セグメントファイルは、ロボットとプログラムコードの位置を含む。本システムは、各ロボットが、そのロボットに属するメインプログラムとセグメントを受信するように、それぞれのロボットコントローラへ、メインプログラムと、セグメントを含むセグメントファイルを自動的に分配するように適合されている。この実施形態により、プレイリストが更に修正しやすくなる。セグメントファイルの読み込みは、ファイルのサイズが大きいため、時間がかかる。この実施形態では、セグメントファイルの読み込みが1回だけで済む。ユーザがプレイリストを修正した場合、メインプログラムのみ、そして任意選択的に新たに追加されたセグメントをロボットコントローラに再読み込みするだけでよい。

【 0 0 6 4 】

プレイリストがロボットにコミットされる前に、互換性チェックを行うことができる。プレイリストに互換性がない場合、ダウンロードが停止する。この機能は、セグメントが互いに互換性があるか、例えば前のセグメントの最後のジョイント値が次のセグメントの最初の接合値と等しいかの互換性チェックを行う。プレイリストに互換性がない場合、以下のメッセージボックス：「プレイリストのセグメントは互換性ありません」が現れ得る。セグメントがOKの場合、以下のダイアログ：「プレイリストはOKです」が示される。

10

20

30

40

50

【 0 0 6 5 】

図 4 に、ディスプレイ 3 2 を保持し、ショーを協調的に演じているディスプレイハンドリングロボットの一例を示す。

【 0 0 6 6 】

以下に、本発明によるシステム及び方法の一例を更に詳しく説明する。システムの基本概念は、ロボットコントローラに送られるシーンのプレイリストを作成することである。各シーンはセグメントを含む。セグメントは、Maya の「ロボットアニメータ」アドインを起点とする。このアドインは、ロボットコード、例えば R A P I D モジュールで書かれたモジュールを生成し、このモジュールは特定のセグメントに対する移動を含む。この移動は、各セグメントの移動は特定の時間に行われるべきであるため、時間係数を用いて絶対ジョイント移動に離散化される。

10

これらはプレイリストハンドラーの基礎である：

1 つの「プレイリスト」がコントローラにダウンロードされる。

「プレイリスト」は 1 - N 「シーン」を含む。

「シーン」は、1 - M 「セグメント」を含み、M はロボットの数である。

「セグメント」は、

- ディスク上の R A P I D モジュールへの絶対経路、
 - R A P I D サブルーチンのコール名（コントローラで実行されるべきサブルーチン）
- 、及び

- セグメント時間、すなわちセグメントに含まれる動きの継続時間
- を含む。

20

【 0 0 6 7 】

シーンごとに 1 つのロボットに対し、セグメントは 1 つのみである。

【 0 0 6 8 】

ユーザインターフェースのボタンをクリックすることによって、プレイリスト、シーン、及びセグメントを編集することが可能なダイアログが示される。ユーザは新たなプレイリストを作成して、既存のプレイリストを修正することができる。「プレイリスト」を選択することによって、含まれているシーンが示される。「シーン」を選択することによって、シーンのセグメントが示される。プレイリストの情報は、Play list handler . exe と同じフォルダに位置する Play list handler Library . xml ファイルに保存される。シーンは幾つかのプレイリストに使用可能である。セグメントは幾つかのシーンに使用可能である。ユーザはプレイリストを追加し、また削除することができる。プレイリストのシーンを編集することができる；新たなシーンを追加することができ、ライブラリから既存のシーンを追加することができ、プレイリストからシーンを削除することができ、シーンの順番を変更することができる。ユーザは、すでに定義されたシーンのライブラリから選ぶことができる。選択されたシーンのセグメントを編集することができる；新たなセグメントを追加することができ、ライブラリから既存のセグメントを追加することができ、プレイリストからセグメントを削除することができ、セグメントの順番を変更することができる。

30

【 0 0 6 9 】

ユーザが「セグメントの追加」をクローズした後、アプリケーションは、セグメントが互いに互換性があるかの互換性チェックを行い、もしない場合には以下のメッセージボックス：「前のセグメントと、このセグメントではジョイント値が違いすぎるため、セグメントを追加できません」が現れる。

40

【 0 0 7 0 】

ライブラリの利用可能なプレイリストが全て示され、ユーザはプレイリストを選択することによってどのプレイリストをロボットコントローラにダウンロードすべきかを選択する。これにより、選択されたプレイリストがロボットコントローラにダウンロードされ、既存のプレイリストが上書きされる。プレイリストをダウンロードする前に、プレイリス

50

トをシミュレーションし、プレイリストによりいかなる衝突又エラーも生じないようにすることが望ましい。

【0071】

ユーザがプレイリストを作成し、それをロボットコントローラにコミットした後で、情報が以下に「DataModule」と称されるRAPIDプログラムモジュールにおいてRAPID変数に書きこまれ、セグメントモジュールが各ロボットコントローラに読み込まれる。オペレータがその後、ショーの次のシーンを実行すると、メインモジュールは「DataModule」のアレイをトラバースして、アレイインデックスに存在するサブルーチン名を呼び出す。

【0072】

プレイリストハンドラーは、プレイリスト記憶装置のファイルに対して作用する。このファイルは、全てのプレイリスト、シーン及びセグメントの記述を含み、新たなプレイリスト構成において使用されるべき既存のアイテムのライブラリとして使用される。ユーザがコントローラにダウンロードされるべきプレイリストを選択すると、プレイリストについての情報がこの記憶装置から読み出される。

【0073】

「DataModule」ファイルは、プレイリスト情報を含み、このファイルは、新たなプレイリストがコントローラにコミットされた時にプレイリストハンドラーによってアップデートされる。ファイルは、オペレータに正確な情報を表示するために、オペレータインターフェースによって読み取られる。

【0074】

プレイリストアレイはシーンを含み、シーンアレイは、そのシーンの第1のインデックスがシーンインデックスであり、第2のインデックスがセグメントインデックスである二次元アレイである。各セグメントに対するロボットの実行時間を含むセグメントタイムアレイも存在する。セグメントタイムアレイは、シーンアレイと同じサイズを有する。

【0075】

以下のシーケンスは、ユーザが新たなプレイリストをコミットした時に使用される。

1. プレイリストハンドラーは、様々な「NumberOfScenes」を設定する。
2. プレイリストハンドラーは、各コンポーネントに「ヌル」を設定することによってプレイリストをクリアする。次に、シーン名がアレイに追加される。2つのシーン（S1及びS2と呼ばれる）がある場合、プレイリストアレイは以下のコンポーネントを有することになる。["S1", "S2", "NULL", "NULL", ...]
3. プレイリストハンドラーは、各コンポーネントに「NULL」を設定することによってシーンをクリアする。次に、シーンのセグメント名がアレイに追加される。2つのシーン（第1のシーンがセグメントSeg1及びSeg2を有し、次のシーンがセグメントSeg3及びSeg4を有する）がある場合、シーンのシーンアレイは以下のコンポーネントを有することになる。["Seg1", "Seg2", "NULL", "NULL", ...], ["Seg3", "Seg4", "NULL", "NULL", ...], ["NULL", "NULL", ...], ...
4. プレイリストハンドラーは、「SegmentTimes」の全てのコンポーネントを最初 - 1 に設定し、その後コンポーネントを正確な時間にアップデートする。Seg1が19sであり、Seg2が5.7であり、Seg3が130.7sであり、Seg4が4.6sである上記と同じ例の結果は以下のアレイである。[19, 5.7, - 1, - 1, ...], [130.7, 4.6, - 1, - 1, ...], [- 1, - 1, ...], [- 1, - 1, ...], ...

【0076】

以下は、上記の例の「DataModule」ファイルの一例である。

10

20

30

40

```

CONST num MAX_SCENES:=20;
CONST num MAX_SEGMENTS:=10;
CONST num joint_tolerance:=0.5;
PERS jointtarget current_joint_pos:=[[0,-49.545,-120,0,79.54,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
PERS jointtarget home_joint_pos:=[[0.000000,-58.000000,58.000000,0.000000,-90.000000,-90.000000],[9E9,9E9,9E9,9E9,9E9,9E9]];

PERS bool PlaySingleScene:=FALSE;
PERS tooldata screenTool:=[TRUE,[[0,0,1],[1,0,0,0]],[500,[0,0,1],[1,0,0,0],0,0,0]];
PERS num SpeedFactor:=0.970299;

PERS string sCurrentPath:="S1";
PERS num nCurrentScene:=1;
PERS num nCurrentSegment:=1;
PERS string sPathSegment:="Seg1";
PERS speeddata vProgSpeed:=[500,500,500,500];
PERS num NumberOfScenes:=1;
PERS num NumberOfSegments:=7;
PERS string SceneList{MAX_SCENES}:=["S1","S2","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL"];
PERS string SceneSegment{MAX_SCENES,MAX_SEGMENTS}:=[["Seg1","Seg1","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL"],
["Seg3","Seg4","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL"],
["NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL"],
["NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL","NULL"]];

PERS num SegmentTime{MAX_SCENES,MAX_SEGMENTS}:=[[19, 5.7,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1],
[130.7,4.6,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1],
[-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1],[-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1],

```

10

【 0 0 7 7 】

メインモジュールはアレイ全体をスキャンし、セグメントの呼び出しにおいて各サブルーチンを実行し、以下はメインモジュールの一例である。

```

FOR i FROM 1 TO NumberOfScenes DO
  IF SceneList{i}<>"NULL" THEN
    nCurrentScene:=i;
    sCurrentPath:=SceneList{i};
    NumberOfSegments:=SegmentCount(nCurrentScene);
    FOR y FROM 1 TO NumberOfSegments DO
      IF SceneSegment{i,y}<>"NULL" THEN
        nCurrentSegment:=y;
        sPathSegment:=SceneSegment{i,y};
        PulseDO\PLength:=0.5,doRobotReadData;
        PreSegment i,y ;
        %sPathSegment %;
        PostSegment i,y ;
        Stop;
      ENDIF
    ENDFOR
  ENDIF
ENDFOR

```

20

30

【 0 0 7 8 】

図5に、本発明の一実施形態によるショーを協調的に演じている複数のロボットの動きを編集し認証するための方法のフロー図を示す。フロー図の各ブロックを、コンピュータプログラムの命令によって実行することができることが理解されるだろう。

【 0 0 7 9 】

本方法は、アニメーションソフトによって定義されるロボットの動きを表す複数のセグメントを取得することによって、各セグメントが、定義された期間の間にロボットの内の1つの動きの記述を含む、取得することを含む(ブロック40)。複数のセグメントは、モーション作成モジュール1からセグメントを受信する又は読み出すことによって取得することができる。セグメントはセグメント記憶装置に保存される。セグメントは、セグメントがどのシーンに属するか、またセグメントがどのロボットに属するかの情報を含む。例えば、セグメントの名前は、セグメントがどのシーンに属するかを含む、又は各セグメントは、セグメントがどのシーンに属するかの情報を含むシーン識別タグを含む。例えば、セグメントの名前は、セグメントがどのロボットに属するかを含む、又は各セグメントは、セグメントがどのロボットに属するかの情報を含むロボット識別タグを含む。複数の

40

50

シーンは、保存されたセグメントに基づいて取得される（ブロック42）。複数のシーンは、モーション作成モジュールからシーンを受信する又は読み出すことによって、あるいは、シーンハンドラー4を用いて保存されたセグメントに基づいてシーンを作成することによって取得することができる。各シーンは、ロボットによって同時に実行されるべき複数のセグメントを含む。シーンについての情報は例えば、前述したようにシーン記憶装置に保存される。

【0080】

シーンのシーケンスを含むプレイリストが取得される（ブロック44）。プレイリストは、システムによってデフォルトのプレイリストとして、あるいはユーザによって作成されうる。ユーザは、シーンを選択して、これらのシーンをシーンのシーケンスに追加することによって新たなプレイリストを作成することができる。プレイリストがユーザインターフェース上に表示され（ブロック46）、ユーザは、シーンのシーケンスを修正することによってプレイリストを編集することが可能になる（ブロック48）。ユーザはプレイリストの編集にユーザインターフェースとやり取りすることにより、プレイリストの修正に関するコマンドを生成する。システムは、修正に関するユーザコマンドを受信する。

【0081】

プレイリストを修正する前に、修正が有効であるかがチェックされる（ブロック50）。修正は、プレイリストの修正部分のシーンに含まれるセグメントのロボットの動きの記述に基づいて認証される。修正は、認証結果に基づいて許可される、あるいは拒否される。これは例えば、前のセグメントの最後の位置値を、プレイリストの修正部分の次に続くセグメントの最初の位置値と比較して、最後の位置値と最初の位置値が等しい、又はその差が限界値よりも小さい場合に修正を許可することによって行われる。修正が有効である場合、ユーザコマンドに基づいてプレイリストが修正される。最後の位置値と最初の位置値との差が限界値よりも大きい場合、修正は有効ではない。修正が有効ではないことが分かった場合、修正は許可されず、ユーザはこれに応じて通知される。認証ステップ52～54はオプションである。例えば、全てのセグメントが同じ開始位置及び終了位置を有するようにセグメントが作成された場合、認証チェックは不必要である。ユーザは、満足のいくプレイリストができるまで、繰り返しプレイリストを編集できる。修正されたプレイリストは、プレイリスト記憶装置、例えば前述したプレイリスト記憶装置7に保存される。

【0082】

ユーザがプレイリストの編集を終了すると、ユーザにはプレイリストに定義されるロボットの動きをシミュレーションすることを選択するオプションがある。あるいは、シミュレーションはオプションではない。プレイリスト、保存されたシーン、及びセグメントに基づいてロボットの動きのシミュレーションが行われる（ブロック56）。プレイリストは、そのシミュレーションに基づいて認証される（ブロック58）。その後、ロボットの動きが衝突するか否かを決定することが可能である。例えば、シミュレーションされたロボットの動きがユーザインターフェース上に表示され、ユーザはロボットの動きが衝突するか否かを視覚的に決定することができる。あるいは、システムがロボットの動きにコリジョンがあるか否かを決定して、ユーザはこれに応じて通知される。別の実施形態では、シミュレーションステップは省略される。

【0083】

図6に、本発明の一実施形態によるショーを協調的に演じている複数のロボットのメインプログラムを生成するための方法のフロー図を示す。ユーザがプレイリストの編集を終了し、プレイリストが認証すると、ユーザはシステムにロボットのメインプログラムの生成を開始するように要求することができる。各ロボットのメインプログラムは、プレイリスト、保存されたシーン及び保存されたセグメントに基づいて生成される。各セグメントは、定義された期間の間にロボットの動きを実行するためのプログラムコードからなるサブルーチンを含む。各サブルーチンはサブルーチン名を有する。プレイリストが読み出される（ブロック60）。メインプログラムは、プレイリストの各シーンについて、ロボッ

10

20

30

40

50

トに属するセグメントのサブルーチン名を読み出すことによって生成される（ブロック 62、64）。各メインプログラムは、ロボットに属するセグメントに含まれるサブルーチン呼び出すシーケンスを含み、サブルーチン呼び出す順番はプレイリストのシーンの順番によって決定される。各ロボットについて、セグメントファイルも生成される（ブロック 66）。セグメントファイルは、ロボットに属するセグメント、又は少なくともロボットに属するセグメントから情報を含む。少なくとも、セグメントファイルは、ロボットの位置と、セグメントのプログラムコードを含む。

【0084】

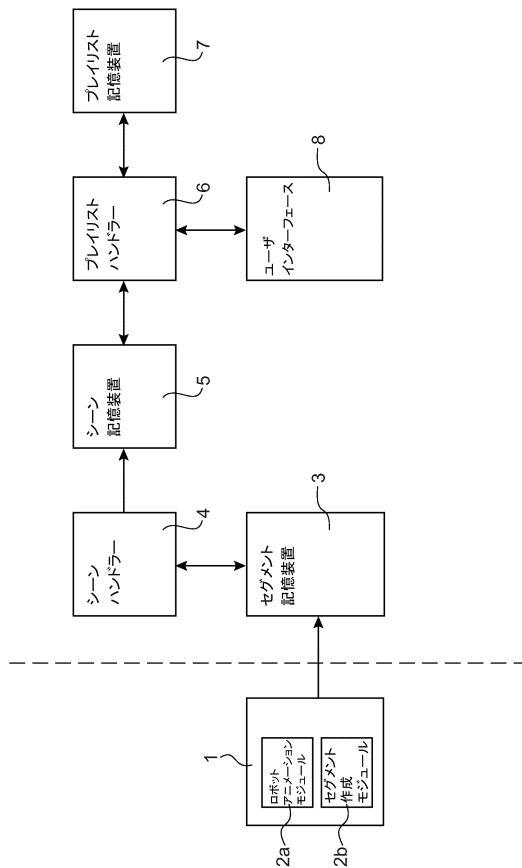
メインプログラムとセグメントファイルが生成されると、ユーザは選択されたロボットコントローラへのメインプログラム及びセグメントファイルの自動的な分配を指図することができる。ユーザからの発令を受信すると、メインプログラム及びセグメントファイルがそれぞれのロボットコントローラに読み込まれる（ブロック 68）。このため、各ロボットは、そのロボットに属するセグメントから関連するメインプログラム及び情報を受信する。シーンは、いかなるオペレータの入力、例えば時間、I/O、イベント又はその他のトリガ等もなしに、他の操作によって開始することができる。

【0085】

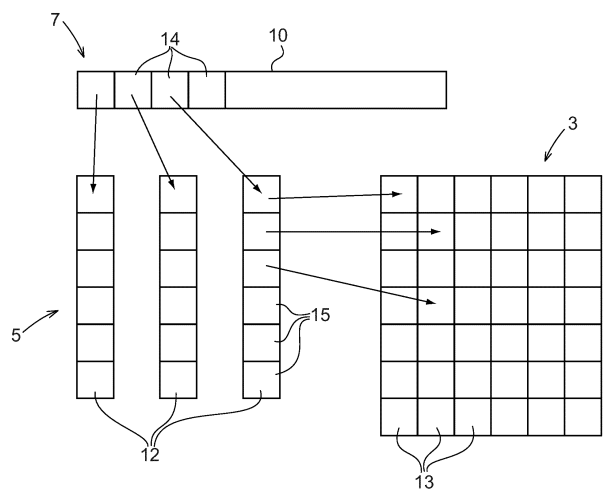
本発明は開示された実施形態に限定されず、以下の特許請求の範囲内で変更及び修正することが可能である。例えば、シーンハンドラー 4 は、モーション生成モジュール 1 に含まれうる。次にモーション生成モジュール 1 は、完成したシーンを直接出力する。

10

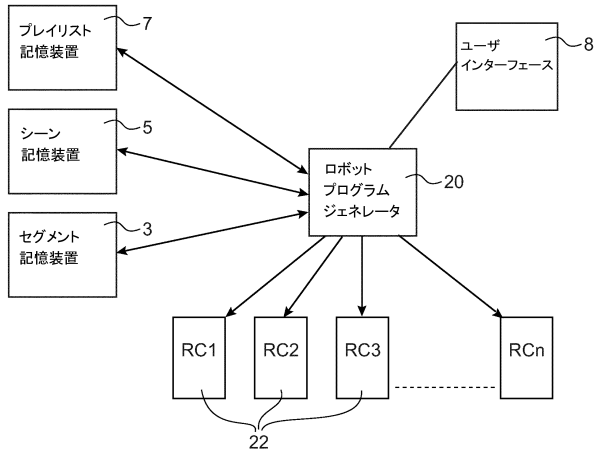
【図 1】



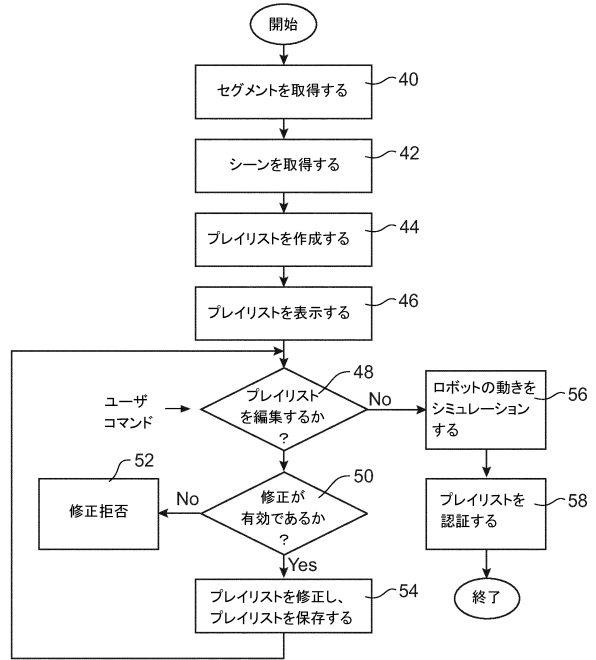
【図 2】



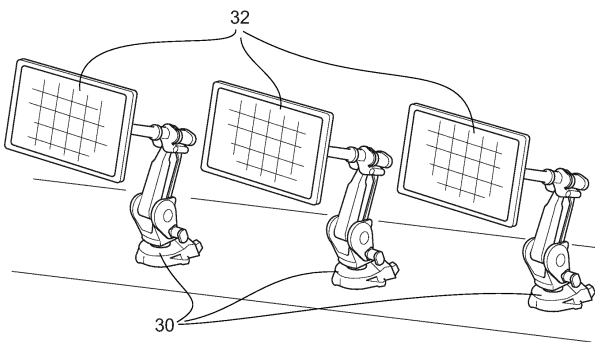
【図3】



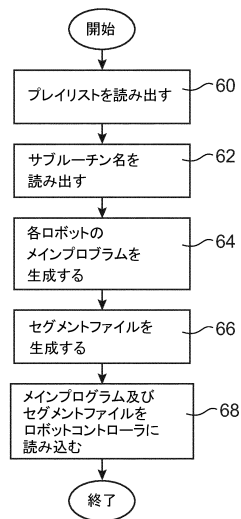
【図5】



【図4】



【図6】



フロントページの続き

審査官 中田 善邦

(56)参考文献 米国特許出願公開第2001/0034559 (US, A1)

特開2006-954 (JP, A)

特開2008-254074 (JP, A)

特開2003-266347 (JP, A)

特開平8-314527 (JP, A)

(58)調査した分野(Int.Cl., DB名)

B25J1/00 - 21/02